

C/C++ Disk G/Ç İşlemleri

Yrd. Doç. Dr. Fehim KÖYLÜ
Erciyes Üniversitesi Bilgisayar Mühendisliği Bölümü

Veriler

- Bitler, byte, sayılar, karakter (char), karakter grubu (char *, string), struct, class, kayıtlar, tablolar, vt
- Dosya karakterleri ve dosya sonu işaretini içerir.

J	a	y	n	e		M	u	r	p	h	y	\n	4	7
	J	o	n	e	s		C	i	r	c	l	e	\n	A
l	m	o	n	d	,		N	C			2	8	7	0
2	\n	<EOF>												

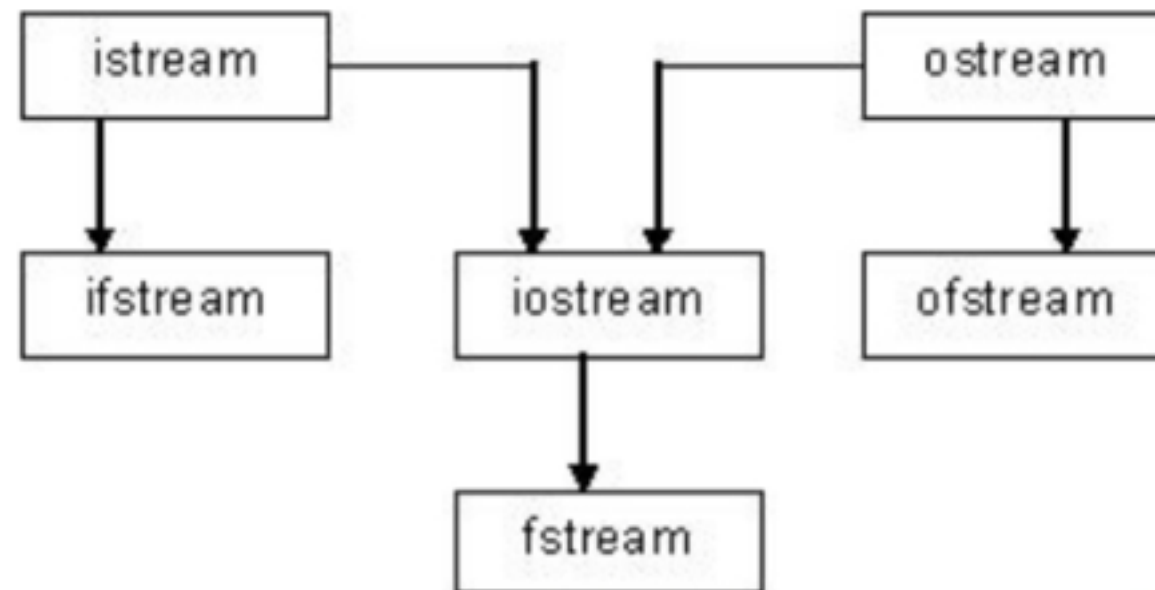
İşlem Adımları

- Dosya adı değişkeni
- Dosya sınıfı ataması
- Dosya aç
- Oku/yaz işlemleri
- Dosya kapat

C++ Sınıfları

Giriş akımı

Çıkış akımı



Dosya Giriş Akımı

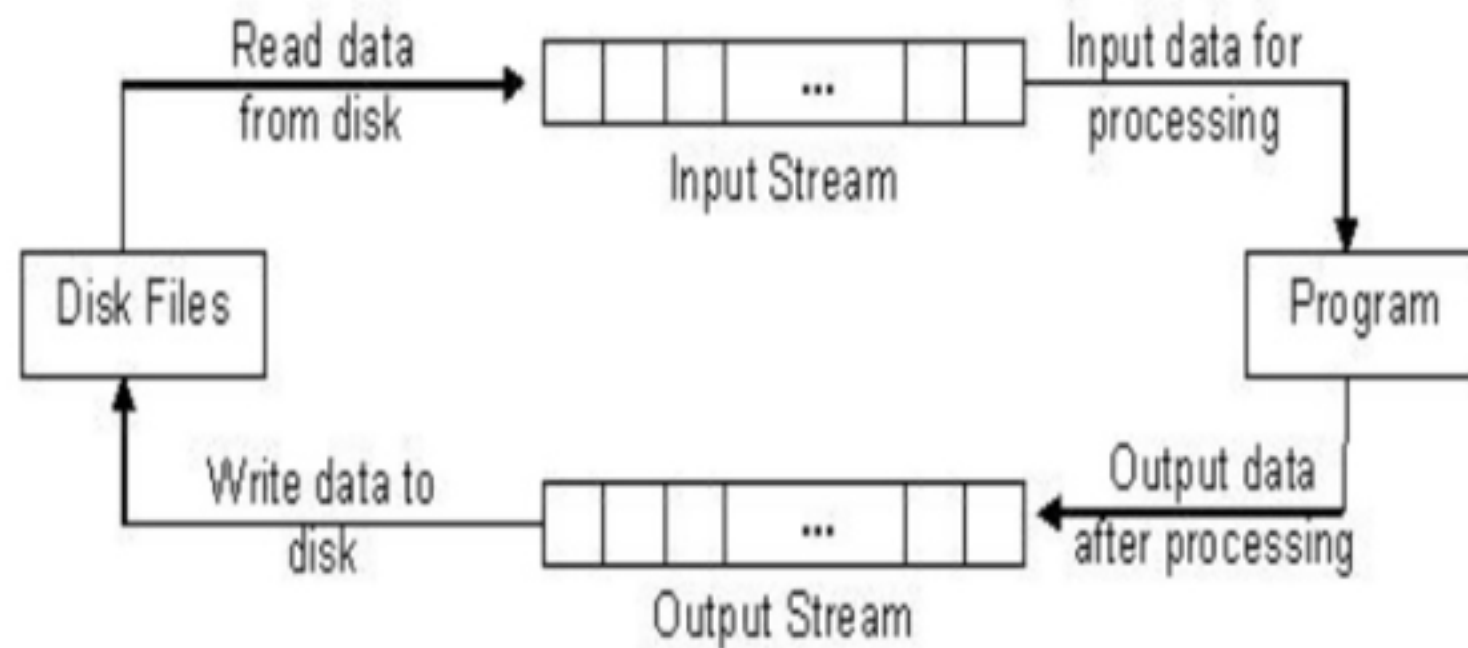
- Dosya üzerinden okuma yapar.
- ifstream : okuma işlemleri için kullanılır.
- fstream : okuma ve yazma işlemleri bir arada yapılabilir.

Dosya Çıkış Akımı

- Dosya üzerine kayıt yapar.
- ofstream : dosya yazma işlemleri için kullanılır.
- fstream : okuma ve yazma işlemleri bir arada yapılabilir.

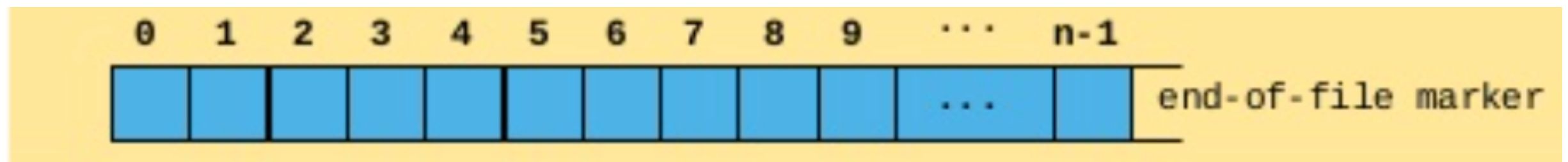
Dosya giriş çıkış akımları

- Disk üzerinde kayıtlı dosya okuma ve yazma işlemleri Giriş ve Çıkış Akımları (Input Stream ve Output Stream) ile yapılmaktadır.



C++ Dosya İşlemi

- C++ için dosyalar byte akımları olarak ele alınır.
- Dosya sonu işareti ile sonlandırma yapılır.
- Ardışık dosya işlemleri
- Rasgele dosya işlemleri



Dosya Açılma Şekilleri

- app: Dosyanın önceki içeriği silinmez. Her ekleme işlemi için dosya sonuna konumlanır ve ekleme yapılır.
- ate: Dosya açıldığında dosya akımı sonuna konumlanır.
- binary: Dosya g/ç işlemi metin biçimi değil ikili biçimde yapılabilir.
- in: Dosya akımı üzerine giriş işlemleri yapılabilir.
- out: Dosya akımı üzerinde çıkış işlemleri yapılabilir. Dosya sıfırlanır.
- trunc: Dosyadaki önceki veri sıfırlanır ve baştan yazılmış olur.

Ardışık Dosya G/Ç

- Ardışık okuma sırasında dosyanın herhangi bir biçimde olması gerekmemektedir. Sırasıyla byte dizileri okunur.
- ifstream, ofstream, fstream
- Yazma işlemi için:
 - ofstream F("dosya adı", fileOpenMode);
 - ofstream F;
F.open("dosya adı",fileOpenMode);
- Okuma işlemi için:
 - ifstream F("dosya adı", fileOpenMode);
 - ifstream F;
F.open("dosya adı",fileOpenMode);

Ardışık Dosya

- `!F` : dosya açarken hata olursa (dosya yok, izin yok, disk dolu vb.) `true` döndürür.
- `if (!F)`
- `F<< değişken; // cout da olduğu gibi değeri dosyaya aktarır.`
- `F.close(); // Dosya işlemi bittikten sonra kapatılır.`

Ardışık Dosya

- Kullanılan kütüphaneler:

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

- Açılma modları:

- ios::out

- ios::in

- ios::ate

- ios::app

- ios::trunc

Ardışık Dosya Yazma Örneği

```
int main(int argc, char const *argv[]) {
    const char* dosya = "test.txt"; // Dosya Adı
    string yazi = "24.11.2016 Perşembe. Bu bir cümledir.";
    const char* karakterDizisi = "Fehim KÖYLÜ";
    int sayisalVeri = 1000000;
    float ondalik = 3.1415;
    ofstream F(dosya, ios::out);
    if(!F){ // Hata olması durumunda dosya açılmayacaktır.
        cout<< "Hata çıktı ve dosya açılmadı";
        return(-1);
    }
    //endl dosya yazma konumunu alt satıra geçirir.
    F<< yazi<<endl; // Dosyaya String değer yazılması
    F<< karakterDizisi<<endl; // Dosyaya Char Dizisi değer yazılması
    F<< sayisalVeri<<endl; // Dosyaya sayısal değer yazılması
    F<< ondalik<<endl; // Dosyaya ondalık sayısal değer yazılması
    // Dosyaya değişkenlerin birlikte yazılması
    F<< yazi<<","<<karakterDizisi<<","<<sayisalVeri<<","<<ondalik<<endl;
    F.close();
    return(0);
}
```

Ardışık Dosya Okuma Örneği

```
int main(int argc, char const *argv[]) {
    const char* dosya = "test.txt"; // Dosya Adı
    string kelime1, kelime2, kelime3, kelime4, kelime5;
    // const char* karakterDizisi = "Fehim KÖYLÜ";
    // int sayisalVeri = 1000000;
    // float ondalik = 3.1415;
    ifstream F(dosya, ios::in);
    if(!F){ // Hata olması durumunda dosya açılmayacaktır.
        cout<< "Hata çıktığı için dosya açılmadı";
        return(-1);
    }
    // Boşluk vee alt satır karakteri görünceye kadar karakterleri
    // değişkene yazar.
    // Bu bir cümledir.24.11.2016 Perşembe
    // kelime1: Bu
    // kelime2: bir
    // kelime3: cümledir.24.11.2016
    // kelime4: Perşembe
    // kelime5: Fehim
    F>>kelime1>>kelime2>>kelime3>>kelime4>>kelime5;
    cout<< kelime1<<kelime2<<kelime3<<kelime4<<kelime5;
    F.close();
    return(0);
}
```

Ardışık Dosya Konumlandırma 1

- Dosya işaretçisi hareketi ile istenilen veri okunur.
- `seekg()` //seek get dosya okuması için
- `seekp()` // seek put dosya yazması için
- Yön:
 - `beg`: akım başlangıcı ile bağlı işaretçi.
 - `cur`: mevcut konum ile bağlı işaretçi hareketi.
 - `end`: dosya sonu ile bağlı işaretçi.
- `F.seekg(0);` // dosya başına konumlanır.

Ardışık Dosya Konumlandırma 2

- `F.seekg(n);` // dosyada baştan itibaren n . bayta konumlanır.
- `F.seekg(n , ios::cur);` //Bulunduğu konumdan n bayt sonraya konumlanır.
- `F.seekg(n , ios::end);` //Sondan geriye n bayt geri konumlanır.
- `F.seekg(0 , ios::cur);` //Sonuncu bayta gider.
- `F.seekp(...)` ; //yukardakiler gibi kullanılır.
- Dosyanın işaretçisinin konumu:
- `int konum = F.tellg();` // okuma işlemi sırasında veya yazma sırasında ise `F.tellp();`

Ardışık Dosya Durum Metotları

- eof() : eofbit bayrağı set edildiği durumda true değeri döndürür. Dosya sonuna geldiğini belirtir.
- fail() : failbit veya hardfail bayraklarından biri set edildiği durumda true değeri döndürür.
- bad() : badbit bayrağı set edildiği durumda true döndürür.
- good() : goodbye bayrağı set edildiğinde true döndürür.
- clear() : parameteresiz kullanıldığında bütün bitleri temizler. Parametre ile kullanılırsa sadece o biti temizler.

Ardışık Dosya Konumlandırma 3

- eof() metodu dosya sonuna geldiğinde true değer döndürür.
- if(inf.eof()){
- inf.close();
- }

```
int main(int argc, char const *argv[]) {  
    const char* dosya = "test.txt"; // Dosya Adı  
    string kelime1;  
    ifstream F(dosya, ios::in);  
    if(!F){  
        // Hata olması durumunda dosya açılmayacaktır.  
        cout<< "Hata çıktığı için dosya açılmadı";  
        return(-1);  
    }  
    F>>kelime1;  
    while (!F.eof()){  
        cout<< kelime1<< "."<<endl;  
        F>>kelime1;  
    }  
    F.close();  
    return(0);  
}
```

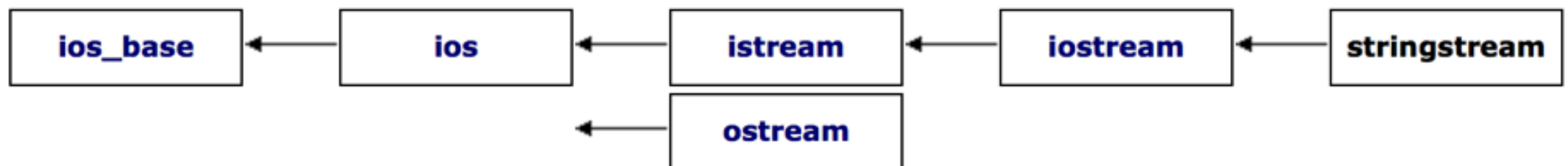
Ardışık Dosya Konumlandırma 4

Her okuma işlemi
boşluk veya \n
karakteri görünceye
kadar gerçekleşir.
Kelime kelime
okunmuş olur.

```
int main(int argc, char const *argv[]) {  
    const char* dosya = "test.txt"; // Dosya Adı  
    string kelime1;  
    ifstream F(dosya, ios::in);  
    if(!F){  
        // Hata olması durumunda dosya açılmayacaktır.  
        cout<< "Hata çıktığı için dosya açılmadı";  
        return(-1);  
    }  
    cout<<"Dosya yeni açıldı! konum:"<<F.tellg()<<endl;  
    F>>kelime1;  
    while (!F.eof()){  
        cout<< kelime1<<"("<<F.tellg()<<")"<<endl;  
        F>>kelime1;  
    }  
  
    cout<<"Dosya sonuna ulaştıktan sonra konum:"<<F.tellg()<<endl;  
    F.clear();  
    F.seekg(0);  
    cout<< "clear() ve seekg() metotları sonrası konum:"<<F.tellg()<<endl;  
    F>>kelime1;  
    while (!F.eof()){  
        cout<< kelime1<<"(2.kez)"<<endl;  
        F>>kelime1;  
    }  
    F.close();  
    return(0);  
}
```

stringstream string akımı

-



Ardışık Dosya Okuma Metotları

- `istream::get(char)` : tek harf oku
- `istream::getline(char* , length, delimiter)` : bir satır oku